GALORATH

Sizing Software From Early Development Artifacts
DoDCAS 2003

Dan Ferens
AFRL/IFEA
daniel.ferens@rl.af.mil

Dan Galorath
Galorath Incorporated
galorath@galorath.com

Project Overview

- **Project awarded to Galorath Incorporated in June 2002 to develop a line of tools for automatically estimating software size from development documents**
  - Awarded by the U.S. Air Force as part of the **Small Business Innovation Research** (SBIR) program
  - **Phase 2** award following a successful **Phase 1**
  - Two-year project, approximately $750,000
  - End product: standalone tools interfacing with industry-leading specification and requirements development tools (Rational Rose, Telelogic DOORS)
- **Requires development of two closely related tools, along with data collection and estimating method enhancement**
  - UML diagrams and structured lists of requirements
  - Will estimate function points, lines of code, etc.

What is Size?

- **“Size” is a measure of software “volume” or functionality**
  - How much code?
  - How many features or functions?
  - Lines, Function Points, Function Based Sizing, Objects, Use Cases, Requirements, and more are viable size metrics
  - Rework is key for sizing system modifications

- **Software size is <u>the main driver</u> of software development effort, cost and schedule -- use the <u>best available</u> estimate of size range**

History of Software Sizing

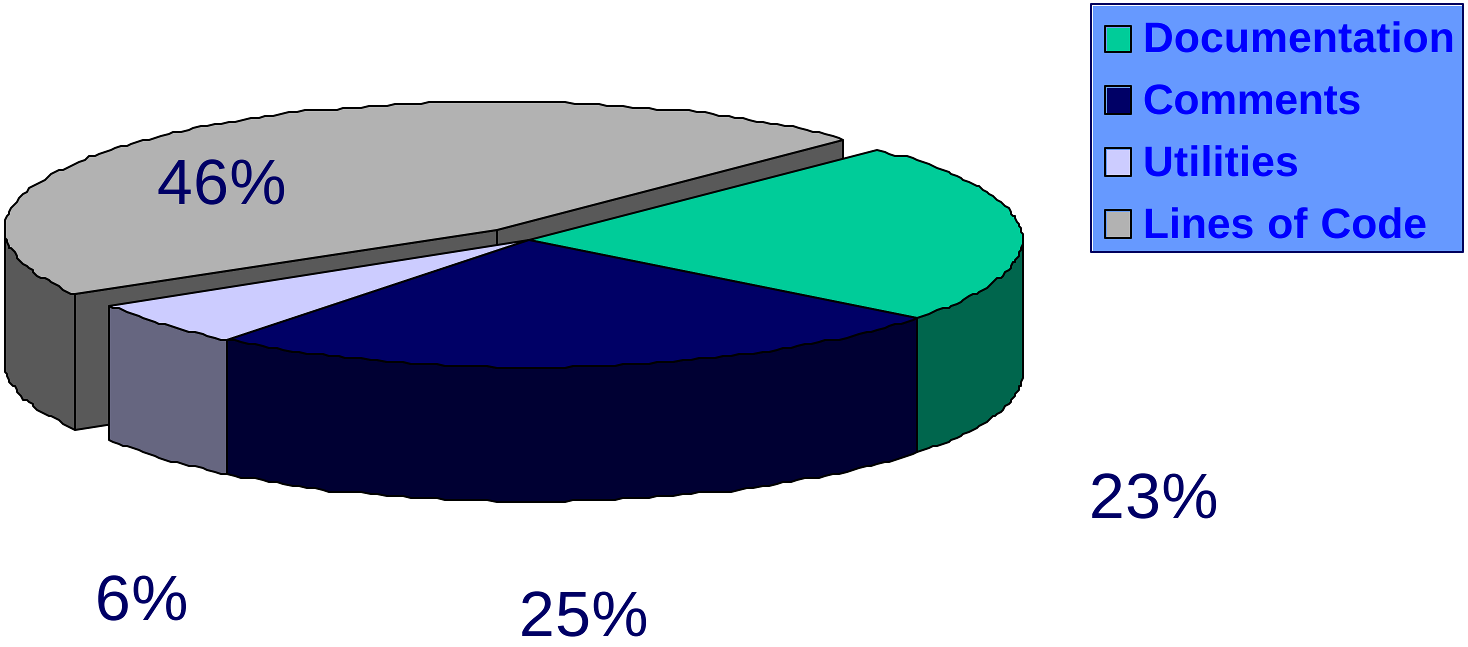
- **Pre 1986 Primitive Methods (E.g. words of memory)**
- **Late 1980’s SLOC and Function Points**
  - Estimating methods available
  - Limitations to size artifacts (SLOC & Function Points)
- **Late 1990’s Object Counts**
  - Several Methods Available
  - Limitations to Object Counts (Different Definitions. Limited Application, New)

Bottom line.. Size still needs research

Therefore this CriticalMass SBIR

Using Proper Line Definition is Important

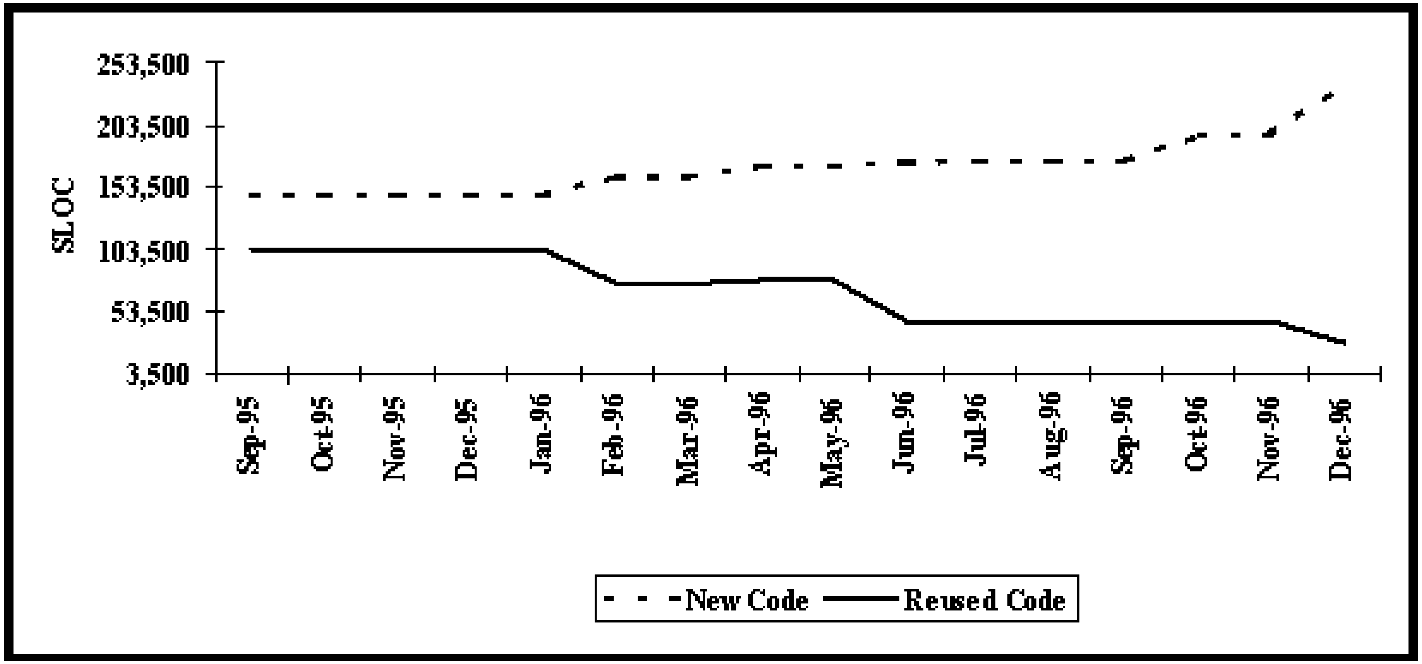
Size estimated at start of development = 1.6 million lines
Actual SLOC was 736,000 lines



Sanitized Actual Program Where Contractor Misstated Size Of Existing Program By Over 2 to 1
Counted Comments, Documentation, Other Items

Size Must Consider Rework of Preexisting Source:

http://fast.faa.gov/pricing/c1919-7.htm#19.7.7



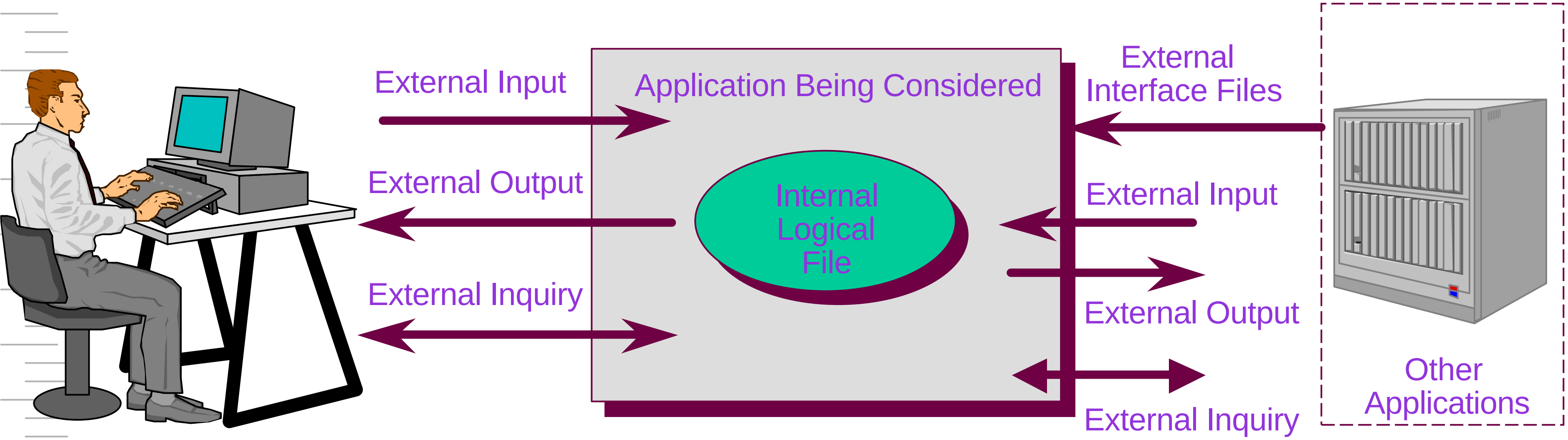
- Chart shows new code growth at the expense of reused code
- Total SLOC grew only 5% during the 17 month period shown
- New code grew from 59% of the total code to 89%.
- Schedule grew about 25%
- **Effective size was far greater than planned**

The Search For The Perfect Size Measure

- **Since we need definitions for software “mass” we need some definition of size**
- **Sizing from the problem space (I.e. Function Points, Use Cases, etc.)**
  - Information potentially available earlier
- **Sizing from the solution space (I.e. lines) estimates a design alternative**
- **Traditional Function Points Work Well In Many Cases But:**
  - The Definitions Are Sometimes Confusing
  - Untrained / Inexperienced People Have Trouble Developing Consistent Function Point Counts
  - Need Special Application For Embedded systems
- **Lines Of Code Works Well In Many Cases But:**
  - Counting Methods Must Count Only Non-Comment Lines
  - Code Generators & Other Modern Development Tools Can Make Lines Of Code Irrelevant
  - New Versus Pre-Existing Must Be Well Understood
  - Line Of Code Counts Can Be Inconsistent based on differences in definitions

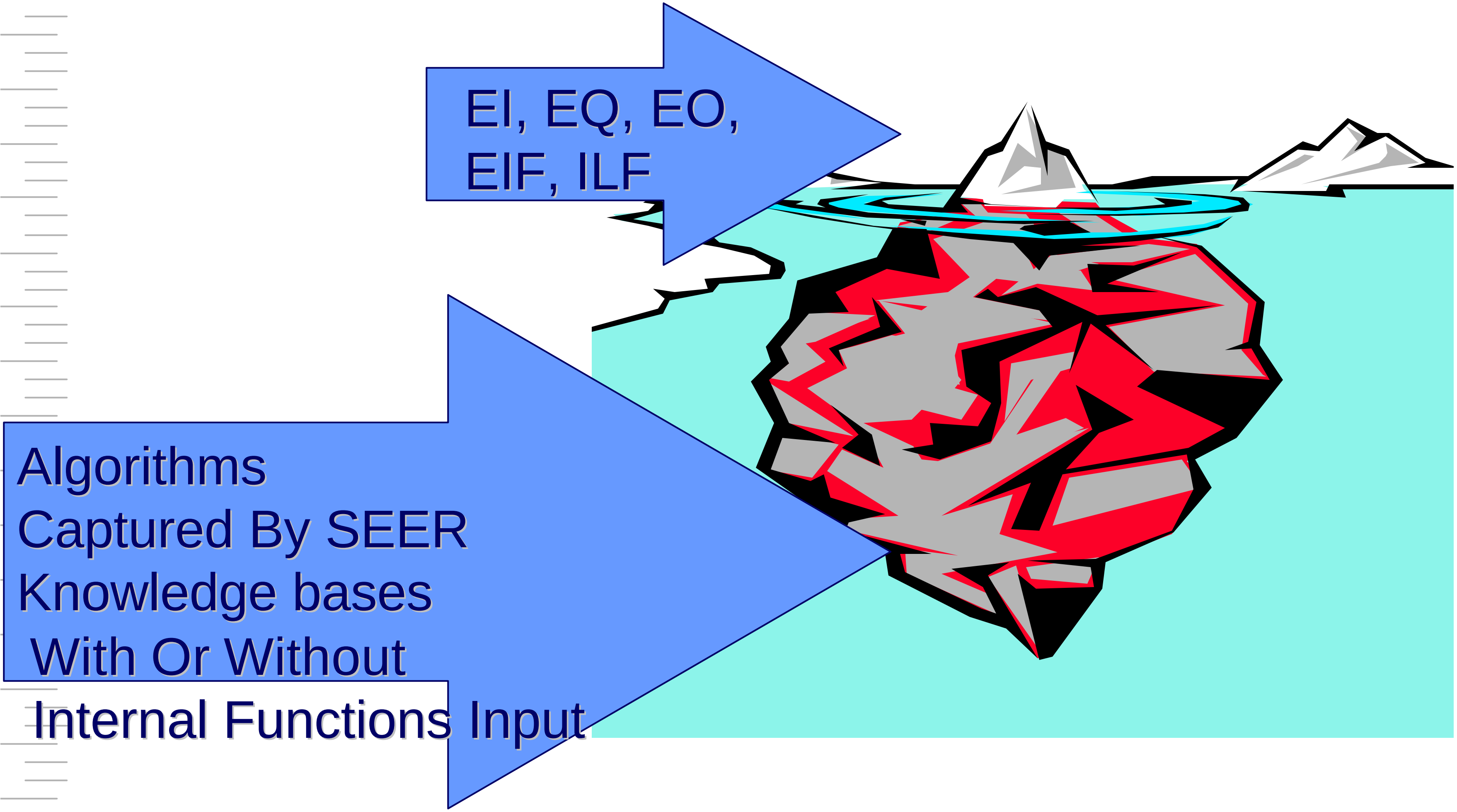
Function Points are a Unit of Measure

Source: IFPUG in a Box



- Functionality as viewed from the user's perspective
- A User is one who writes the system requirements, not just a software operator

For Compute Intensive Systems Traditional IFPUG Function Points Uncover Part Of The Effort



EI, EQ, EO, EIF, ILF

Algorithms Captured By SEER Knowledge bases With Or Without Internal Functions Input

Some Function Point Miscounting Observed During Our Research

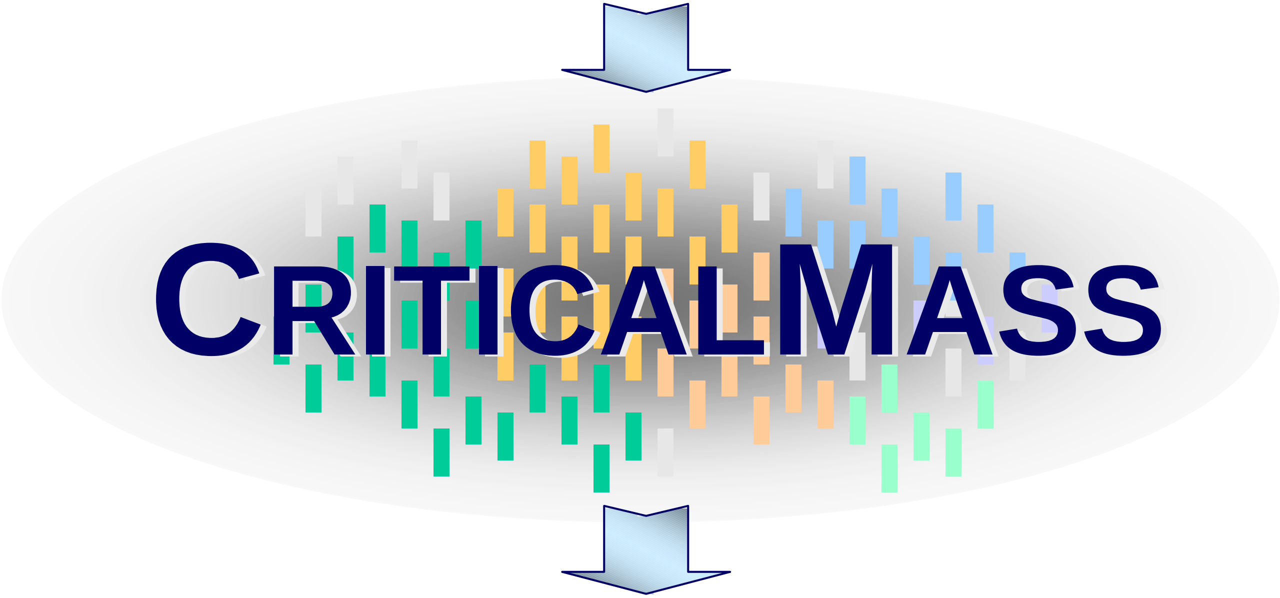
- Programmers Over Estimate (or Over Count Existing Systems) "Get Credit" for Their Work
- Inflated Counts For Reengineered Systems Due To “Forgotten” Functionality (Typically Up To 20% In Long Lived Legacy Systems)
- Different Counters May Count Function Points Very Differently Depending on Their Perception of the User Perception (Over 70% Difference With 2 Experienced Counters)
- Difficulty Describing Entirely Internal Functions (Outside The Automated Information System Domain)

An Automated Sizing Toolset

From Software Design & Requirements Artifacts…

Use Case Models (UML)

Structured Requirements Documents



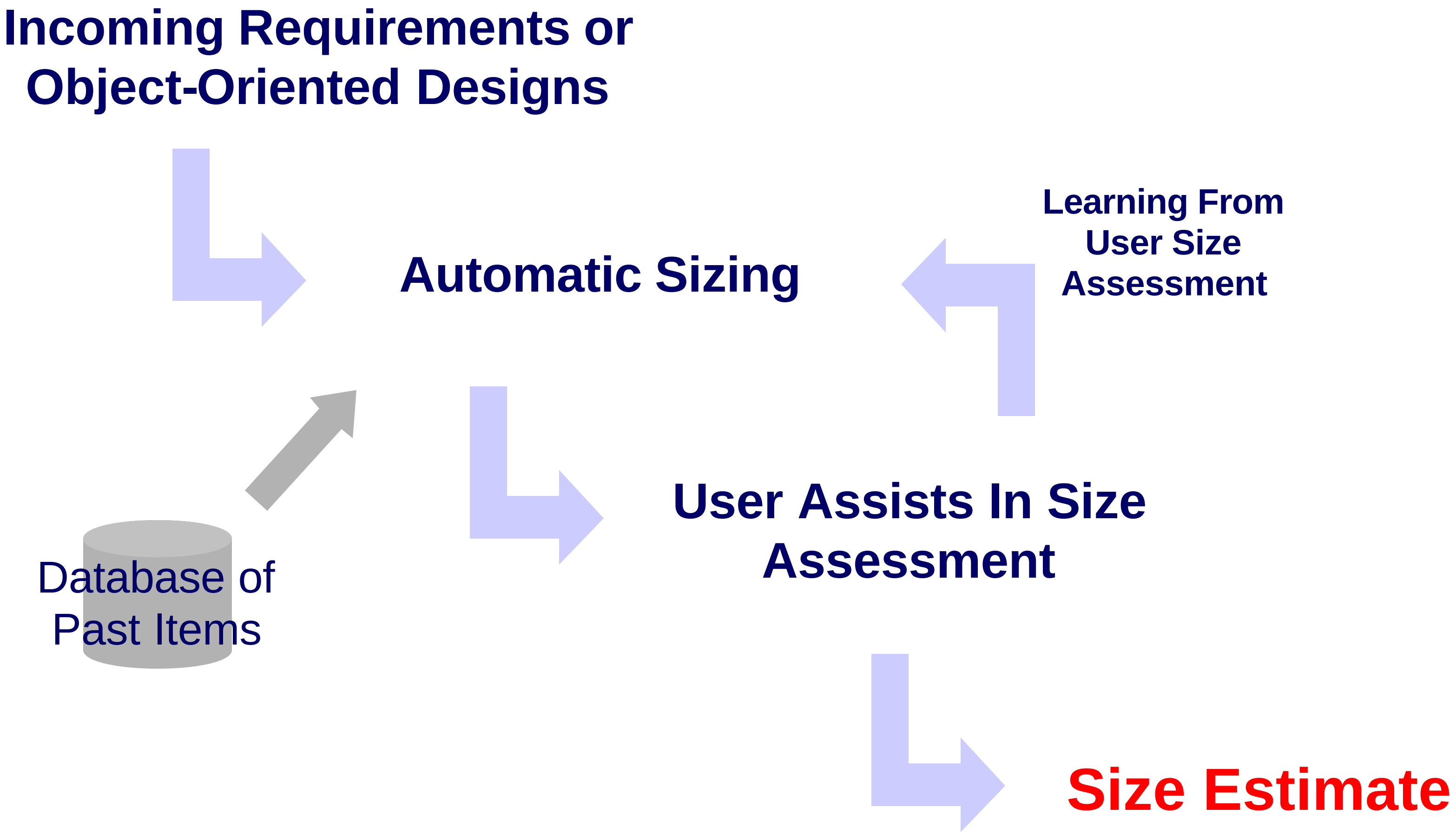
…A Size Estimate Is Produced

Use With Productivity Metrics

Reports

Other Tools

Work Flow



Incoming Requirements or Object-Oriented Designs

Automatic Sizing

Learning From User Size Assessment

Database of Past Items

User Assists In Size Assessment

Size Estimate

Apply Size Estimation Methodology
(Source: Galorath Size Estimation Methodology Guidance)

Evaluate All Sources of Software Size…

New Size

Pre-existing Size (rework)

Generated Code

COTS/GOTS

Glue Code

Integrated Code

Multiple Size Estimates

Total Size Estimates	**Least**	**Likely**	**Most**
Expert Judgement	**12000**	15500	17000
Relevant Range by Analogy	19850	24750	32540
Sizing Database	~~**8000**~~		**46000**
Functional Analysis		27540	
SEER-SSM	15450	**22650**	29850
Delphi Analysis	16788	19750	22713
Composite	**12000**	**22650**	**46000**

Viable Size Range

Expert Judgment

Functional Analysis

SEER-SSM Analysis

Counts for Pre-existing

Sizing Databases

Analogies

…Using Multiple Methods

GALORATH

SBIR

Galorath Observations From Reviews

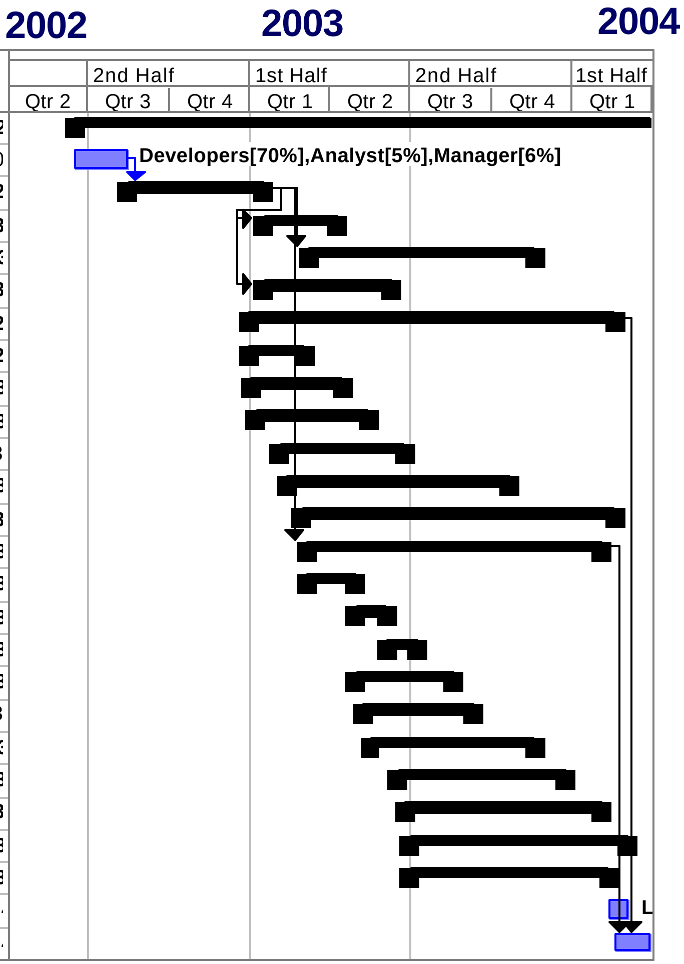
Size Mistake	Consequence
1. Don’t Spend Sufficient Time In Software Sizing	Size Estimates that don’t reflect the program Programs overrunning cost / schedule estimates
2. Don’t Use Clear Definitions Of Size	Size Measures Are Unreliable for Cost / Schedule Estimates
3. Don’t Consider Size Growth in Their estimates OR reduce size estimates to achieve desired cost	Optimistic cost / schedule Programs overrunning cost / schedule estimates
4. Ignore Historical Sizes As Basis For Analogy Due To Differences In Language and Methodology	Lost opportunity to forecast future better from past

Recent US Mil/Aero Sizing Growth Studies

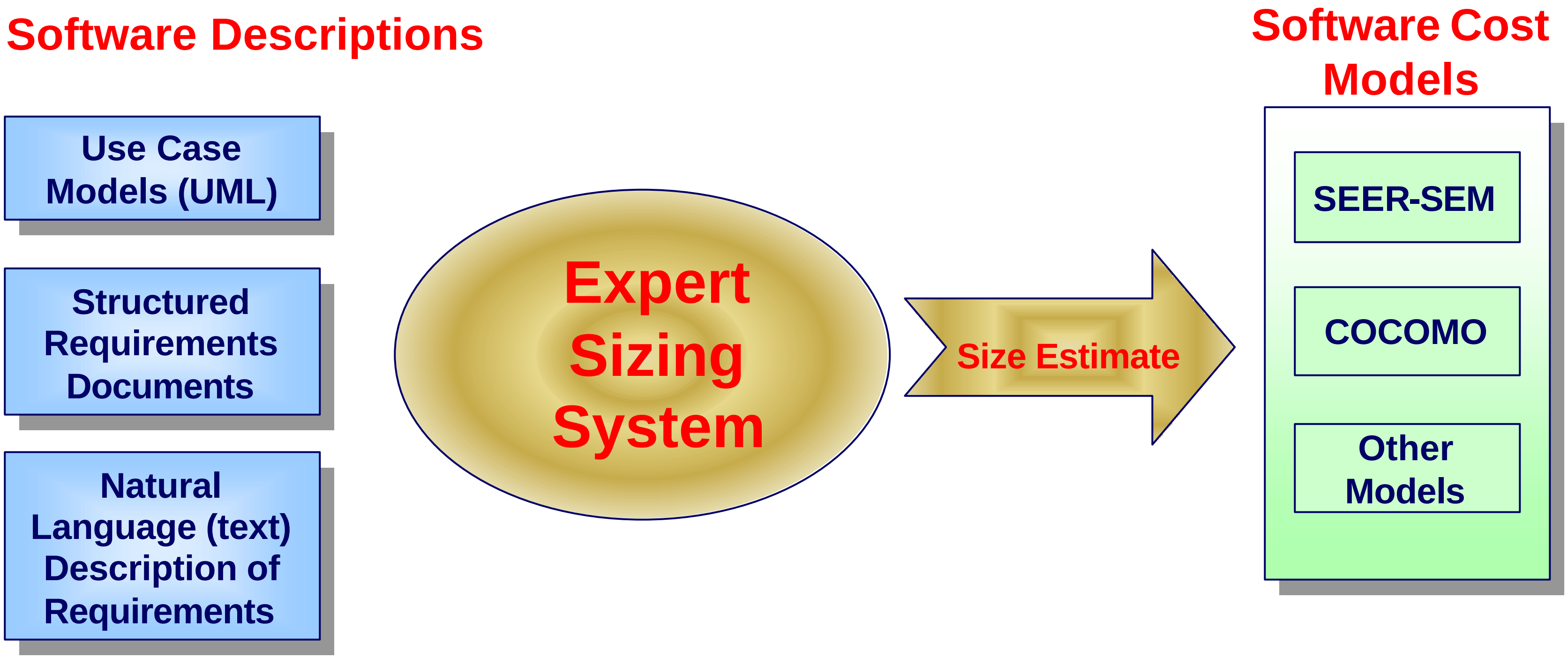
- USAF / ASC 100% Plus Circa 1996
- OSD Size Growth Study 43% Growth From Government Size Estimates
- NCAA Size Growth Study 22% Size Growth

Project Plan

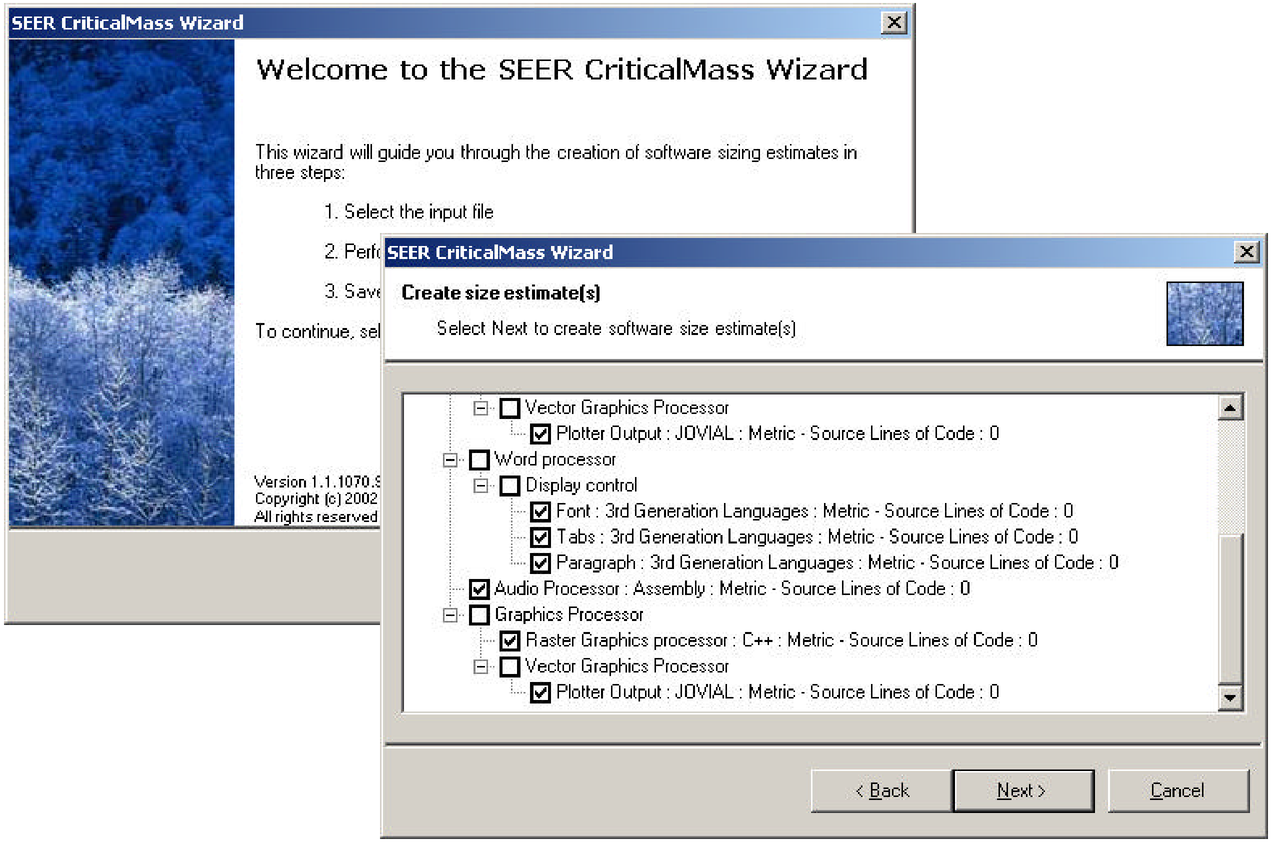
ID	Task Name	Duration	Work	Start
1	**CriticalMass**	**467.27 days?**	**872.83 days**	**Mon 6/17/02**
2	Inception	43.43 days	35.28 days	Mon 6/17/0
3	**Framework & Prototype**	**110 days**	**150 days**	**Thu 8/15/02**
8	**Framework (DOM)**	**59.07 days?**	**15 days**	**Thu 1/16/03**
12	**Size by Comparison Recast**	**183 days?**	**80 days**	**Fri 3/7/03**
16	**Generic repository interface**	**103.84 days?**	**31 days**	**Thu 1/16/03**
21	**Rose**	**296.5 days?**	**232.6 days**	**Tue 12/31/02**
22	**Rose Data Extraction**	**46 days?**	**46 days**	**Tue 12/31/02**
27	**Rose derived metrics**	**75.5 days?**	**18.3 days**	**Wed 1/1/03**
33	**Rose auto sizing with static factors**	**90.5 days?**	**40.8 days**	**Wed 1/8/03**
39	**Rose sizing validation**	**101.5 days?**	**38.5 days**	**Tue 2/4/03**
45	**Rose sizing calibration**	**180.5 days?**	**55 days**	**Wed 2/12/03**
51	**Rose Feedback loop**	**253.73 days?**	**34 days**	**Thu 2/27/03**
57	**DOORS**	**239.5 days?**	**266.95 days**	**Wed 3/5/03**
58	**DOORS derived metrics, iteration 1**	**39.5 days?**	**23.95 days**	**Wed 3/5/03**
62	**DOORS data extraction (2nd iteration)**	**25 days?**	**25 days**	**Wed 4/30/03**
66	**DOORS derived metrics, iteration 2**	**25 days?**	**25 days**	**Wed 6/4/03**
70	**DOORS auto sizing with static factors**	**80 days?**	**36 days**	**Wed 4/30/03**
76	**DOORS sizing validation**	**89 days?**	**21 days**	**Thu 5/8/03**
82	**DOORS derived metrics, iteration 3**	**133 days?**	**71 days**	**Fri 5/16/03**
88	**DOORS sizing calibration**	**137 days?**	**31 days**	**Mon 6/16/03**
94	**DOORS Feedback loop**	**161 days?**	**34 days**	**Tue 6/24/03**
100	**Derived metrics repository**	**177 days?**	**22 days**	**Mon 6/30/03**
101	**Elaboration**	**162 days?**	**7 days**	**Mon 6/30/03**
105	Construction	15 days?	15 days	Wed 2/11/0
106	Transition	29.77 days	40 days	Wed 2/18/0



Newest Sizing Techniques



User View – Select Items From WBS



SEER CriticalMass Wizard

Welcome to the SEER CriticalMass Wizard

This wizard will guide you through the creation of software sizing estimates in three steps:

1. Select the input file
2. Perfo
3. Save

To continue, sel

Version 1.1.1070.9
Copyright (c) 2002
All rights reserved

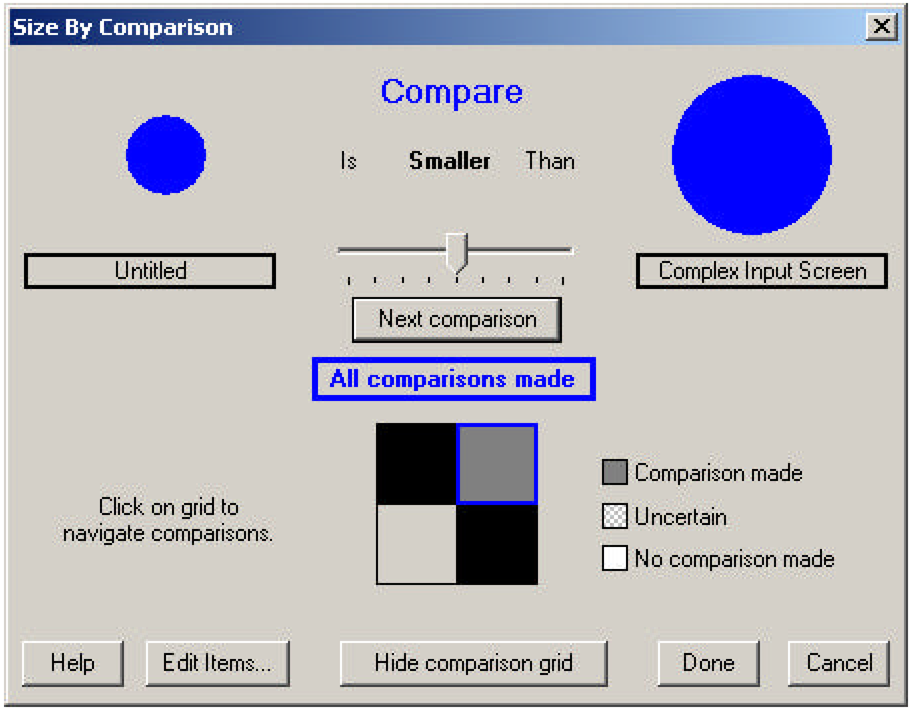
SEER CriticalMass Wizard

Create size estimate(s)

Select Next to create software size estimate(s)

- □ Vector Graphics Processor
  - ☑ Plotter Output : JOVIAL : Metric - Source Lines of Code : 0
- □ Word processor
  - □ Display control
    - ☑ Font : 3rd Generation Languages : Metric - Source Lines of Code : 0
    - ☑ Tabs : 3rd Generation Languages : Metric - Source Lines of Code : 0
    - ☑ Paragraph : 3rd Generation Languages : Metric - Source Lines of Code : 0
- ☑ Audio Processor : Assembly : Metric - Source Lines of Code : 0
- □ Graphics Processor
  - ☑ Raster Graphics processor : C++ : Metric - Source Lines of Code : 0
  - □ Vector Graphics Processor
    - ☑ Plotter Output : JOVIAL : Metric - Source Lines of Code : 0

User View – Size Items By Comparison



Size By Comparison

Compare

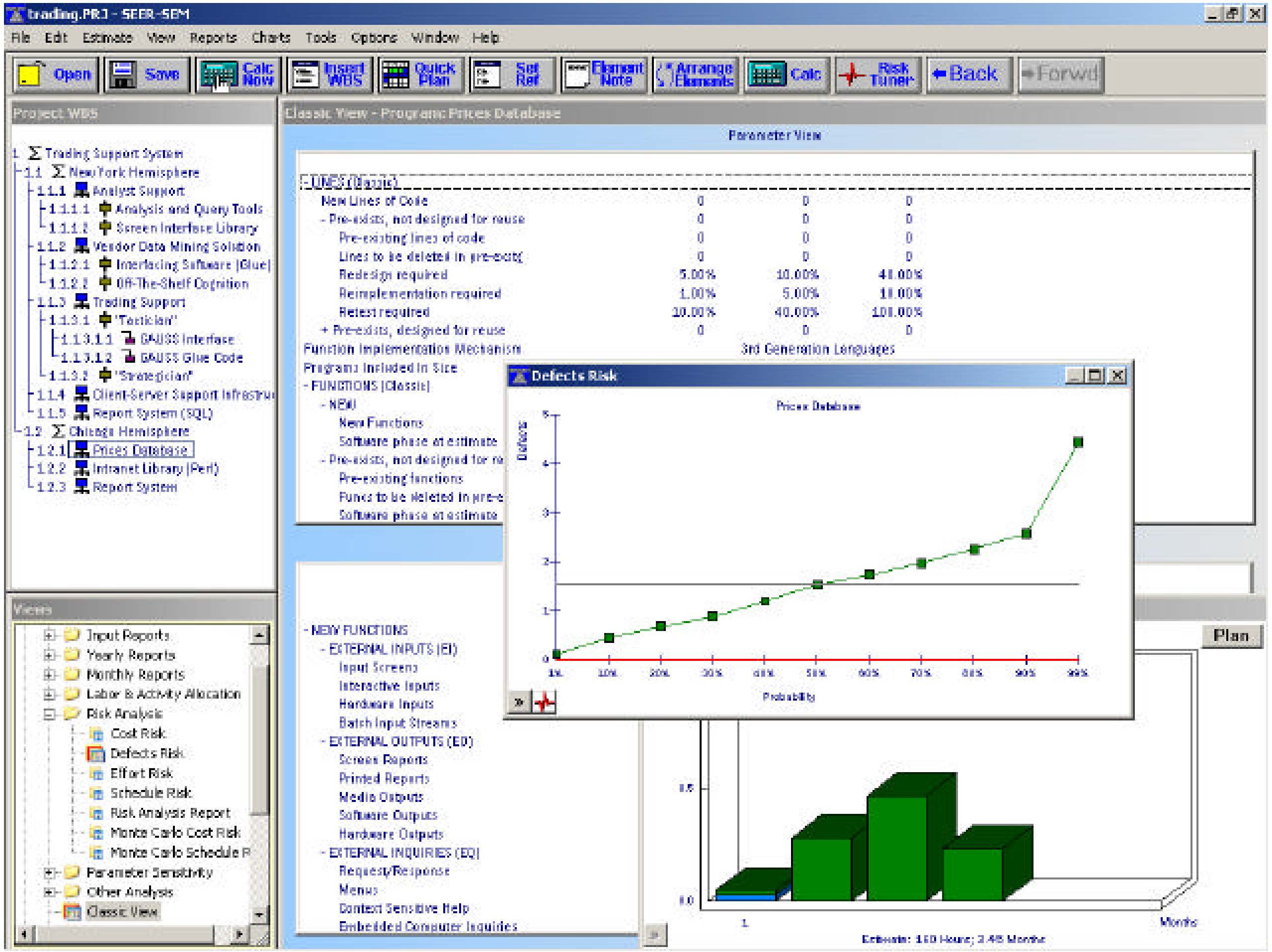
Untitled Is Smaller Than Complex Input Screen

All comparisons made

Click on grid to navigate comparisons.

- ▨ Comparison made
- ▩ Uncertain
- □ No comparison made

Potential Next Step – Output To Software Estimating Tool



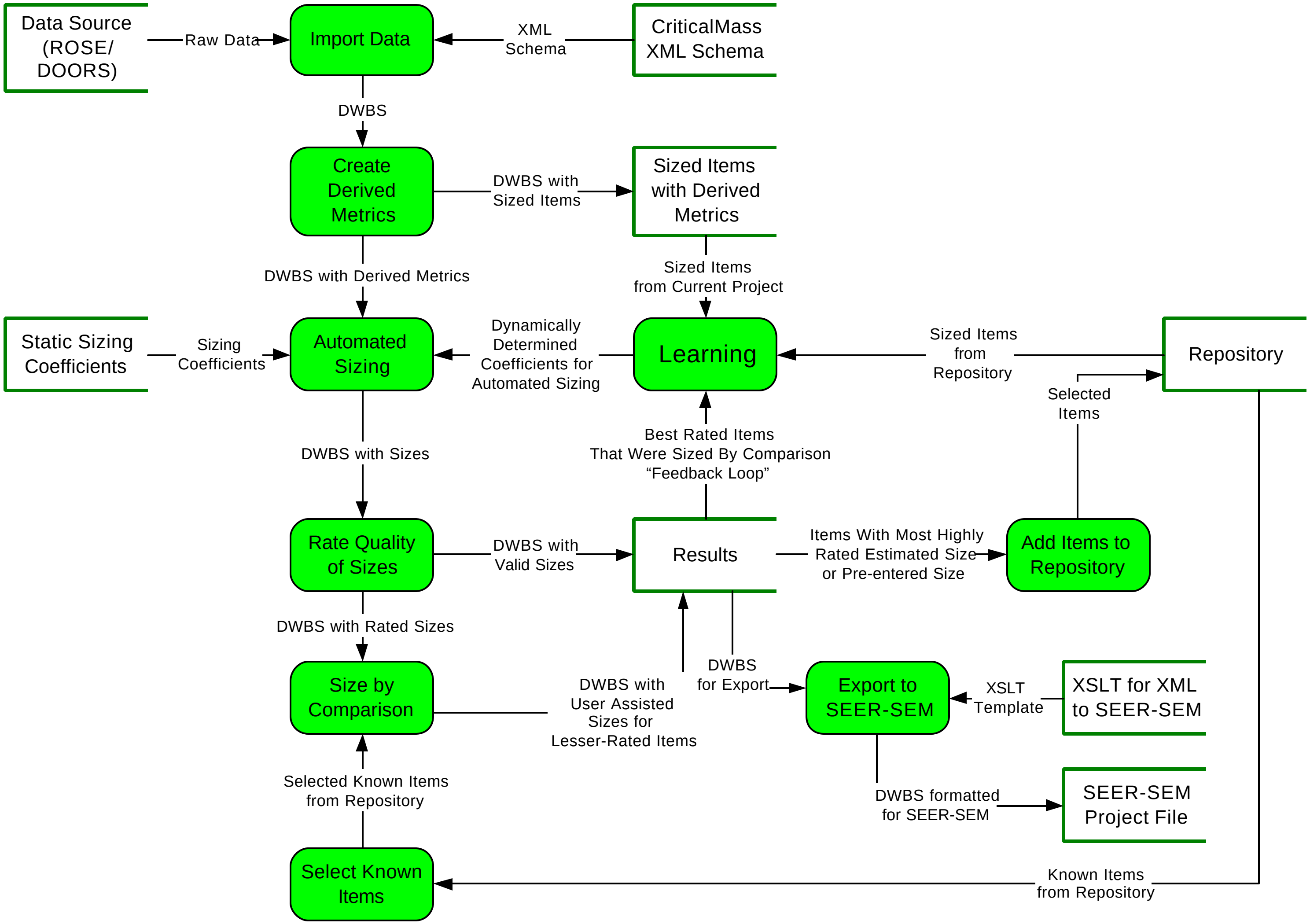
1 Σ Trading Support System
1.1 Σ New York Hemisphere
1.1.1 Analyst Support
1.1.1.1 Analysis and Query Tools
1.1.1.2 Screen Interface Library
1.1.2 Vendor Data Mining Solution
1.1.2.1 Interfacing Software (Glue)
1.1.2.2 Off-The-Shelf Cognition
1.1.3 Trading Support
1.1.3.1 'Tactician'
1.1.3.1.1 GAUSS Interface
1.1.3.1.2 GAUSS Glue Code
1.1.3.2 'Strategician'
1.1.4 Client-Server Support Infrastru
1.1.5 Report System (SQL)
1.2 Σ Chicago Hemisphere
1.2.1 Prices Database
1.2.2 Intranet Library (Perl)
1.2.3 Report System

Input Reports
Yearly Reports
Monthly Reports
Labor & Activity Allocation
Risk Analysis
Cost Risk
Defects Risk
Effort Risk
Schedule Risk
Risk Analysis Report
Monte Carlo Cost Risk
Monte Carlo Schedule R
Parameter Sensitivity
Other Analysis
Classic View

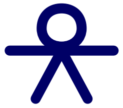
- LINES (Classic)			
New Lines of Code	0	0	0
- Pre-exists, not designed for reuse	0	0	0
Pre-existing lines of code	0	0	0
Lines to be deleted in pre-exist	0	0	0
Redesign required	5.00%	10.00%	40.00%
Reimplementation required	1.00%	5.00%	10.00%
Retest required	10.00%	40.00%	100.00%
+ Pre-exists, designed for reuse	0	0	0
Function Implementation Mechanism		3rd Generation Languages	
Programs Included In Size			
- FUNCTIONS (Classic)			
- NEW			
New Functions			
Software phase at estimate			
- Pre-exists, not designed for re			
Pre-existing functions			
Funcs to be deleted in pre-e			
Software phase at estimate			

- NEW FUNCTIONS
- EXTERNAL INPUTS (EI)
Input Screens
Interactive Inputs
Hardware Inputs
Batch Input Streams
- EXTERNAL OUTPUTS (EO)
Screen Reports
Printed Reports
Media Outputs
Software Outputs
Hardware Outputs
- EXTERNAL INQUIRIES (EQ)
Request/Response
Menus
Context Sensitive Help
Embedded Computer Inquiries

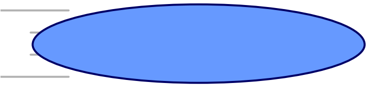
Data Flow Diagram


Data Source (ROSE/ DOORS)
Raw Data
Import Data
XML Schema
CriticalMass XML Schema
DWBS
Create Derived Metrics
DWBS with Sized Items
Sized Items with Derived Metrics
DWBS with Derived Metrics
Sized Items from Current Project
Static Sizing Coefficients
Sizing Coefficients
Automated Sizing
Dynamically Determined Coefficients for Automated Sizing
Learning
Sized Items from Repository
Repository
Selected Items
DWBS with Sizes
Best Rated Items That Were Sized By Comparison “Feedback Loop”
Rate Quality of Sizes
DWBS with Valid Sizes
Results
Items With Most Highly Rated Estimated Size or Pre-entered Size
Add Items to Repository
DWBS with Rated Sizes
Size by Comparison
DWBS with User Assisted Sizes for Lesser-Rated Items
DWBS for Export
Export to SEER-SEM
XSLT Template
XSLT for XML to SEER-SEM
DWBS formatted for SEER-SEM
SEER-SEM Project File
Selected Known Items from Repository
Select Known Items
Known Items from Repository

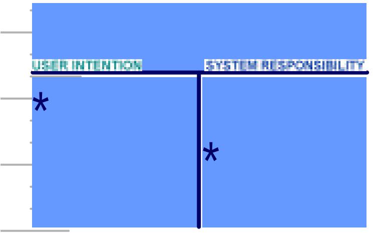
UML Increasingly Popular System Definition Approach



- **An actor is anything that interfaces with your system (I.E. people, other software, hardware devces, data stores, networks, etc.)**
  - Actors exert outside influence over which our system has no control
  - Each actor defines a particular role

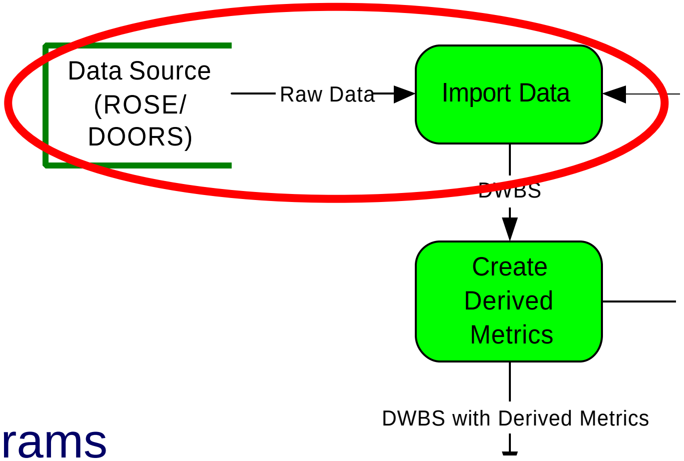


- **A Use case is a behavior of the system that produces a measurable result of value to the actor**
  - Describes things actors want the system to do
  - Use Cases diagrams can be decomposed to increasingly simpler ones until one use case per actor or per use case
    - – "Packages" are the containers for use cases



- **Use cases are elaborated via text (details fleshed out)**

Types of Data

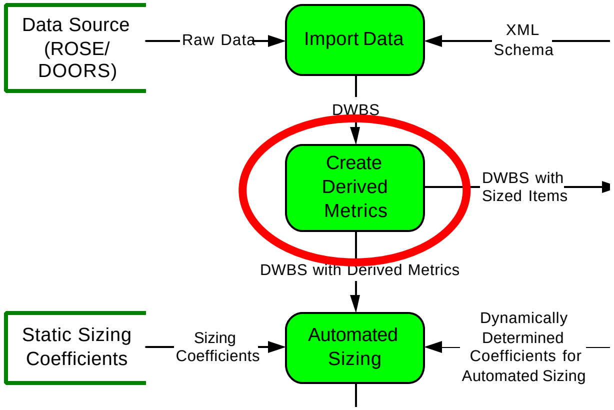


- **UML artifacts**
  - Use Cases
  - Static diagrams
  - State diagrams
  - Deployment and package diagrams

- **Requirements repositories**
  - Initially from DOORS
  - Potentially diverse types of information within actual requirements
  - Schema is deliverable WBS, other structures depending on user input

Create Derived Metrics - UML

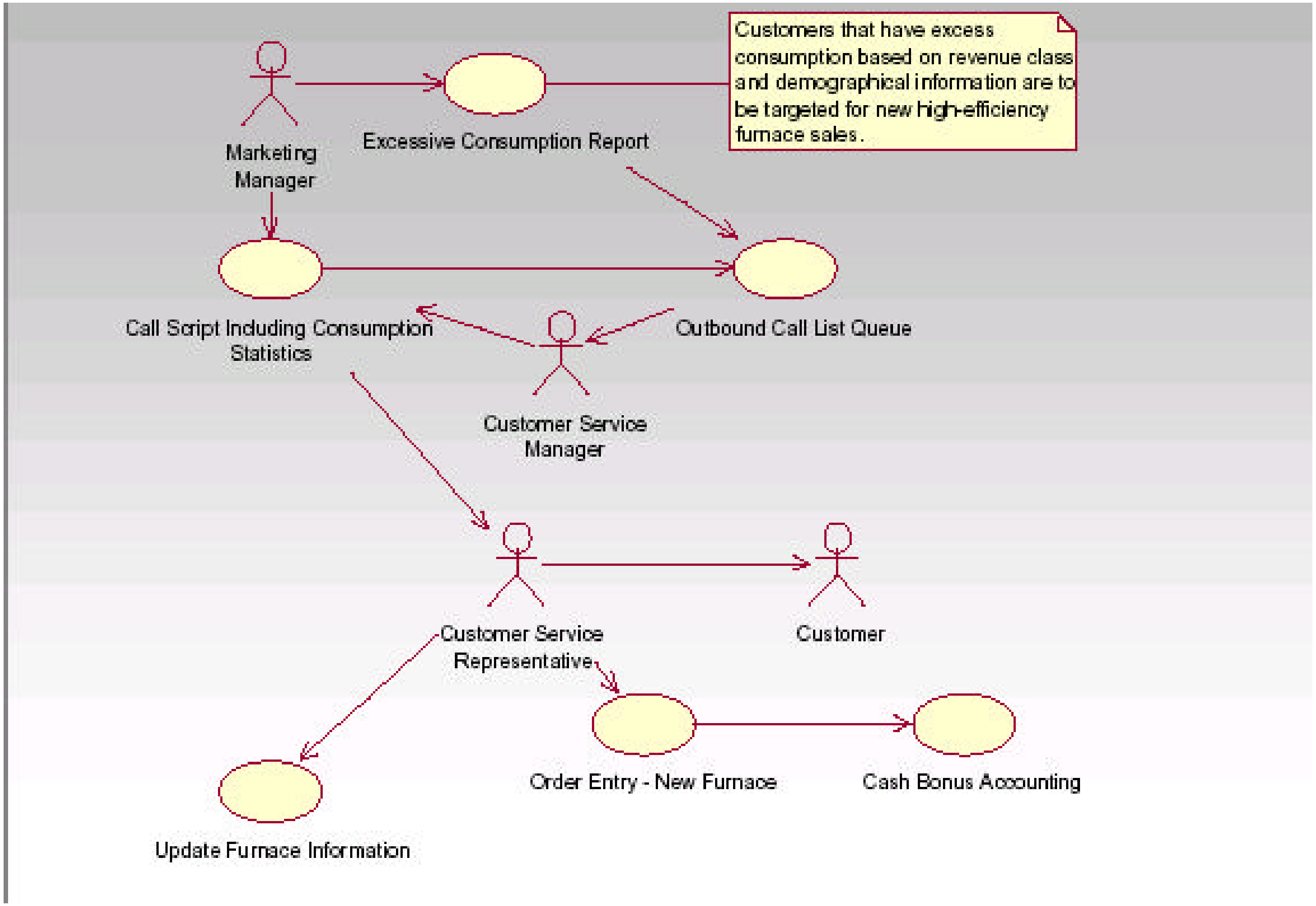
- **For the total system or system component to be sized**
  - Total number of Use Cases
  - Total number of Use Case – Actor relationships
  - Total number of Use Case – Use Case relationships
  - Total number of Classes
  - Total number of Attributes
  - Total number of Operations
  - Total number of Associations


Data Source (ROSE/ DOORS)
Raw Data
Import Data
XML Schema
DWBS
Create Derived Metrics
DWBS with Sized Items
DWBS with Derived Metrics
Static Sizing Coefficients
Sizing Coefficients
Automated Sizing
Dynamically Determined Coefficients for Automated Sizing

- **For each Use Case**
  - Number of related Actors
  - Number related Use Cases

- **For each Class**
  - Number of Attributes
  - Number of Operations
  - Number of Associations (possibly broken down by Multiplicity and Navigability)
  - Number of States
  - Number of State Transitions (arcs between states in State Diagram)

Example Use Case Diagram


Marketing Manager
Excessive Consumption Report
Call Script Including Consumption Statistics
Outbound Call List Queue
Customer Service Manager
Customer Service Representative
Customer
Order Entry - New Furnace
Cash Bonus Accounting
Update Furnace Information

Customers that have excess consumption based on revenue class and demographical information are to be targeted for new high-efficiency furnace sales.

Use Case Elaboration (Documentation)

This use case will describe the steps required to run Norton Disk Doctor. The purpose for running this is as follows:

Norton Disk Doctor diagnoses and repairs a variety of disk problems. It performs several tests, checking everything from the disk's partition table to its physical surface. If Norton Disk Doctor finds a problem, it notifies you before making repairs. If you check Automatically Fix Errors, Norton Disk Doctor makes the necessary repairs automatically.

After diagnosing and repairing a disk, Norton Disk Doctor displays an easy-to-read report that lists the problems found, the problems fixed, and the areas of the disk that checked out okay.

1. Actor(s)

1.1 IT Support Clerk

2. Flow of Events

2.1 Basic Flow

2.1.1 IT Support Clerk selects Diagnose.

2.1.2 *System examines disk for errors*

2.1.3 *System displays results*

2.1.4 IT Support Clerk confirms results

2.1.5 End of Use Case.

3. Alternative Flows

3.1 Continuing from 2.1.2 - System identifies errors on the disk

3.1.1 *System identifies errors on the disk and displays fix option*

3.1.2 IT Support Clerk chooses to correct errors

3.1.3 System corrects errors and displays results.

3.1.4 End of use case

3.2 Continuing from 2.1.2 - System identifies errors on the disk

3.2.1 *System identifies errors on the disk and displays fix option*

3.2.2 IT Support Clerk chooses not to fix errors on disk

3.2.3 System skips fix and displays results.

3.2.3 *E*nd of Use Case

4. Special Requirements

5. Pre-Conditions

5.1 System navigated from Norton SystemWorks to Norton Utilities to Norton Disk Doctor

5.2 Norton Disk Doctors is correctly installed on PC

6. Post Condition

6.1 Norton Disk Doctor closed and System returns to idle condition

Use Case Estimation Funded by AFRL (Partial findings)

- **Count Use Case Points (Original Concept Gustav Karner Objectory AB, part of Rational Software)**
- **Amplification & Enhancements by Galorath (Dr. Denton Tarbet & Lee Fischman)**
  - Divide use cases into simple, medium, and difficult based on characteristics of the number of actors and the actions for each case.
  - Linear combination of weighted counts

Adjusted Correlation Coefficient (R^2) = 0.984802

- **Next ran methodology with Lockheed actuals**

Galorath Analysts Achieved better than 20% effort by this SBIR's methodology

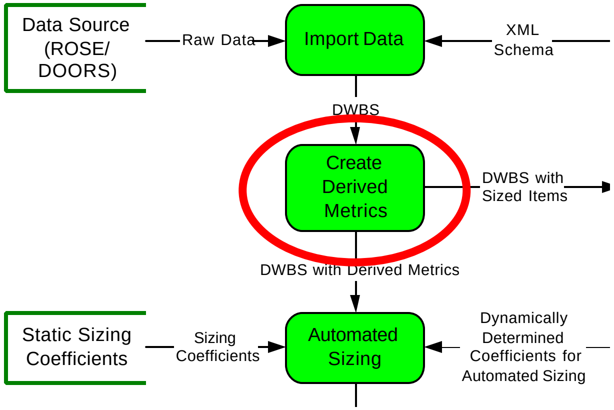
Lockheed Analysts Achieved Better than 12% accuracy (more visibility into use case complexity)

Size/effort Based on Use Case Model

- **Use case models appear to provide a way to derive early size estimates *for domain specific applications*.**
  - Conceptual architecture is expressed in model
  - Use Cases documented with UML
  - UML has an approved standard
  - Industry is considering use cases for estimation (Project estimation, verification of requirements, generation of test cases)
- **Specified use case models for 5 domain-specific programs**
- **Developing a size estimation model from use case artifacts**
- **During Phase II, will test on additional programs from the domain to validate the initial results**

Create Derived Metrics - DOORS

- **Number of requirements linked with WBS item**
- **Word count**
- **Information density (using compression algorithms)**
- **Source documents in modules (Word, Excel, other)**
- **General key words**
  - “Shall”, “screen”, “database”, etc.
- **Context-specific nouns**
  - “Sensor”, “pilot”, “APU”, etc.
- **Grammatical constructs**
- **Page artifacts**
  - Bullets, lines, pictures, etc.
- **Document length**



Data Source (ROSE/ DOORS)
Raw Data
Import Data
XML Schema
DWBS
Create Derived Metrics
DWBS with Sized Items
DWBS with Derived Metrics
Static Sizing Coefficients
Sizing Coefficients
Automated Sizing
Dynamically Determined Coefficients for Automated Sizing

Note overlap between derived metrics in DOORS and those carried in textual documents.

Like Function Points, Use Cases Must be defined Properly

1. The system boundary is undefined or inconstant.
2. The use cases are written from the system's (not the actors') point of view.
3. The actor names are inconsistent.
4. There are too many use cases.
5. The use-case specifications are too long.
6. The use-case specifications are confusing.
7. The use case doesn't correctly describe functional entitlement.
8. The customer doesn't understand the use cases.
9. The use cases are never finished.
10. Use Cases are at inconsistent levels

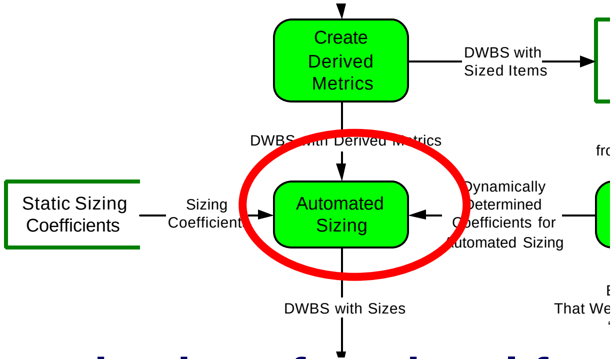
From Lilly, S., Use Case Pitfalls: Top 10 Problems from Real Projects Using Use Cases, Proceedings of TOOLS USA '99, IEEE Computer Society, 1999.

Automated Sizing

"Mining Data For Size Relationships"

Pre-specified functional forms ('y = ax + b' , 'log y = ax + cx2 + c', etc.) **+ estimating methods to obtain coefficients** (what is a? b?) **=**

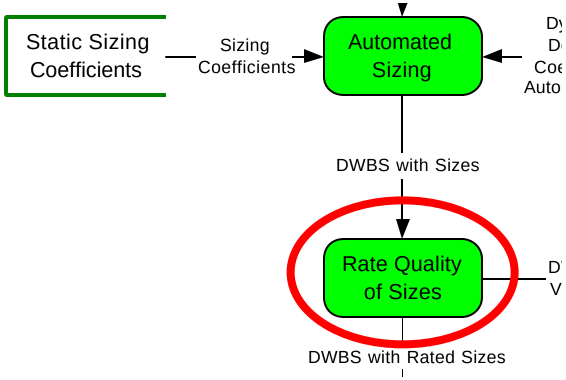
dynamically learning size estimating


Create Derived Metrics
DWBS with Sized Items
DWBS with Derived Metrics
fro
Static Sizing Coefficients
Sizing Coefficients
Automated Sizing
Dynamically Determined Coefficients for Automated Sizing
DWBS with Sizes
That We

- **Analysts will determine best functional forms beforehand**
- **Separate sets of functions for Rose (UML) and DOORS (repository)**
- **At least one functional form for Rose & DOORS each, usually more**
- **Functions' coefficients will be estimated dynamically**

Rate Quality of Sizing

Did automatic sizing do a good job?

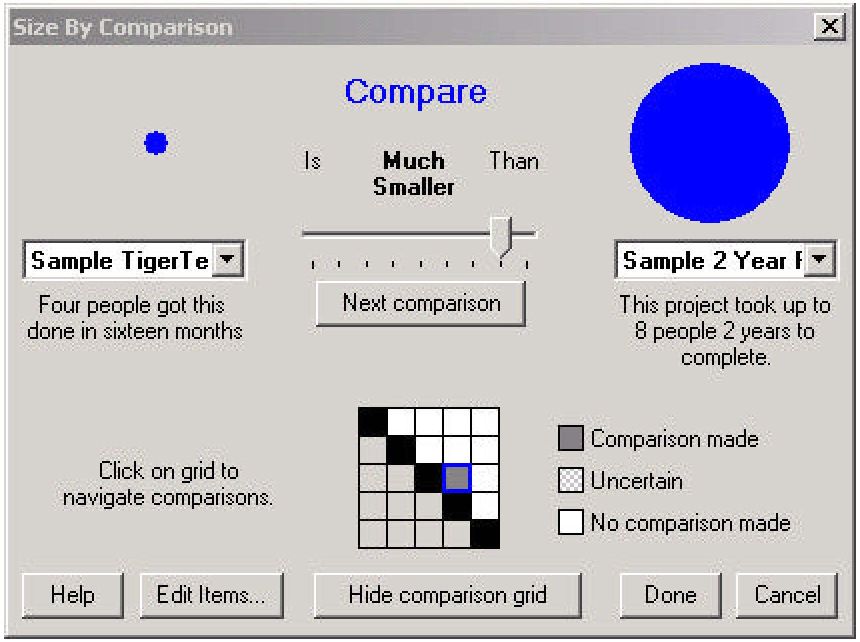


Criteria for rating an estimate:

- **How much did the estimate change, given the estimating function used? There will ideally be multiple passes each time using different data and different functional forms.**
- **Did the estimate rely on low- or high-confidence indicators (# of use cases vs. # of classes)?**
- **What is the statistical confidence level of the coefficients be used?**

Relative Sizing

- **Relative estimation performs well and will be a part of Galorath's methodology**



Size By Comparison

Compare

Is **Much Smaller** Than

Sample TigerTe

Four people got this done in sixteen months

Sample 2 Year F

This project took up to 8 people 2 years to complete.

Click on grid to navigate comparisons.

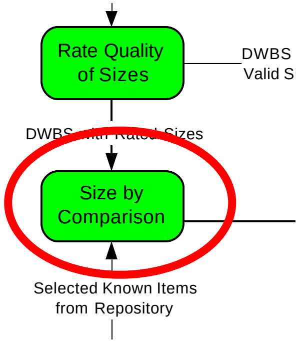
■ Comparison made

▨ Uncertain

□ No comparison made

Size By Comparison

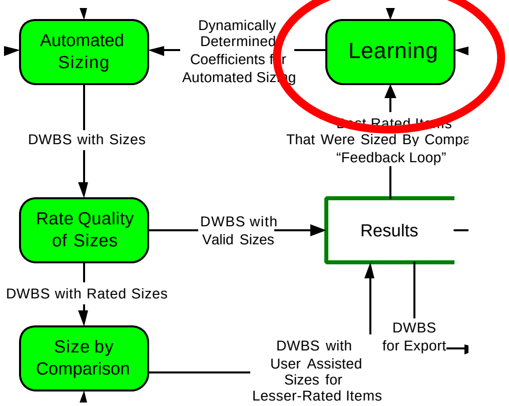
Augments Automated Sizing



- Lets the user provide input about least understood items
- Entries are made via the user-friendly method of pair-wise comparisons
- A special implementation of SBC is being made to fit into the CriticalMass framework
- Uses the proven Analytic Hierarchy Process (AHP) algorithm

Learn From Known Sizes and User Input

Statistical Methods For Determining Coefficient Values



Automated Sizing

Dynamically Determined Coefficients for Automated Sizing

Learning

DWBS with Sizes

Best Rated Items That Were Sized By Compa

"Feedback Loop"

Rate Quality of Sizes

DWBS with Valid Sizes

Results

DWBS with Rated Sizes

DWBS for Export

Size by Comparison

DWBS with User Assisted Sizes for Lesser-Rated Items

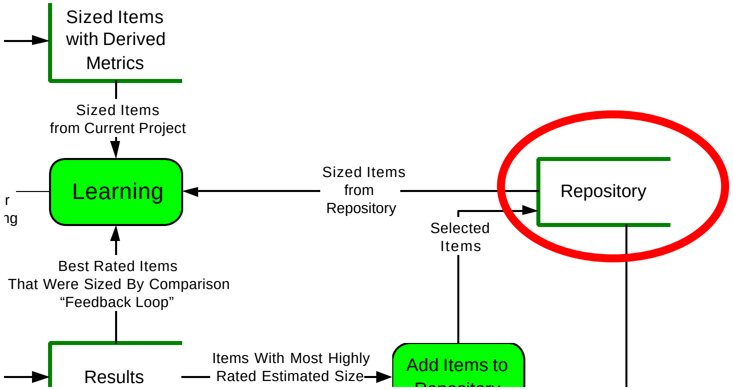
- Inputs are sized items from the repository, existing project, or items sized with user help

- Statistical methods to be included will depend on technical need. They may include Least Average Deviation, Least Squares Regression, etc.

- Many of the methods are being developed for other projects and will be reused at very little additional cost

Repository

A Database of Sized Items Described By Derived Metrics


Sized Items with Derived Metrics
Sized Items from Current Project
Learning
Sized Items from Repository
Repository
Selected Items
Best Rated Items That Were Sized By Comparison "Feedback Loop"
Results
Items With Most Highly Rated Estimated Size
Add Items to

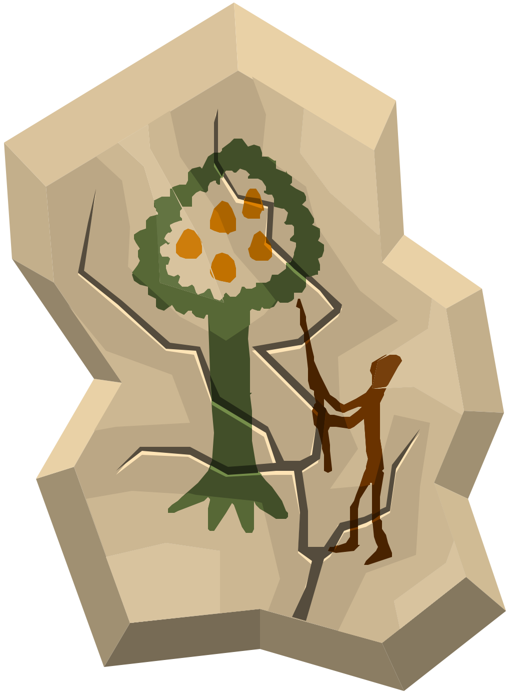
- **Database will contain records formatted similarly to this:**

  {Name, Description, Level, Knowledge bases, Metric, Size, DerivedMetric1, DerivedMetric2, etc.}

- **Items most frequently at CSC and CSU level**
- **Items might not contain the full potential set of derived metrics**
- **The repository will grow with customer use; certain comparison-sized items may be added so that it is 'trained' based partially on user input**

Data Gathering

- **Data gathering is a key project activity**
- **We are pursuing internal & external sources**
  - **Internally generated data**—Extracting information from our projects, synthesizing data (use cases, etc) to validate aspects of the tools
  - **External data**—Obtained from industry and agency partners.
- **Data collection risk mitigation**
  - We will try to make headway by establishing close industry partnerships with strong incentives to participate
  - Access to unusual data sources such as accounting records
  - When necessary, we will synthesize data under laboratory conditions to verify that the tools work



GALORATH

Air Force SBIR

Key Points

- The system automatically builds lists of items to be sized
- Sizes are determined, as much as possible, automatically
- The system learns with use, improving estimates with more data
- CriticalMass encourages collaboration on scoping tasks: groups of subject matter experts can combine their assessments

Data Collection Activities

- The Air Force is eliciting additional software data collection
- Will benefit the entire community
- To participate contact:

Dan Ferens
ferensd@rl.af.mil
(315) 330-4098